N.37-38 LUGLIO/AGOSTO '88

L. 8.000-Frs. 12,00

fare ELETTRONICA

Realizzazioni pratiche • TV Service • Radiantistica • Computer hardware

REALIZZAZIONI PRATICHE

Wattmetro Hi-Fi

Allarme telefonico

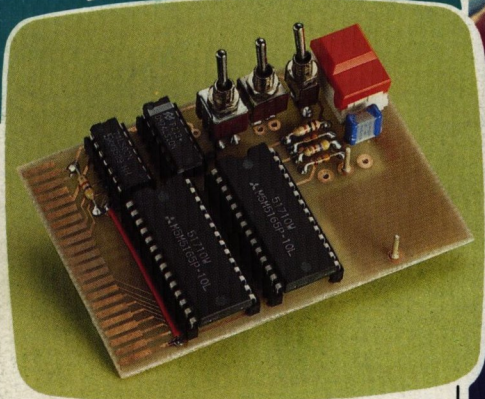
COMPUTER HARDWARE

Telefono automatico col C64

ROM Fittizia per il C64

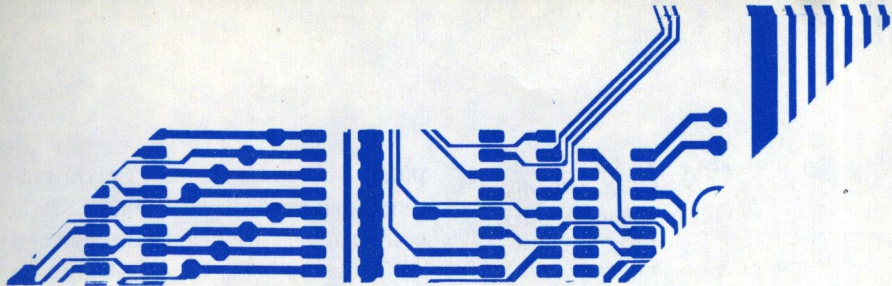
SERRATURA A COMBINAZIONE

Taxe Percue (Tassa Riscossa) Milano Ferr. Corr.
Spedizione in Abb. Post. Gruppo III/70

IN COLLABORAZIONE CON ETI ELECTRONICS TODAY INTERNATIONAL

RADIANTISTICA

Rosmetro

TV SERVICE

Metz 6983

ROM FITTIZIA PER IL C64

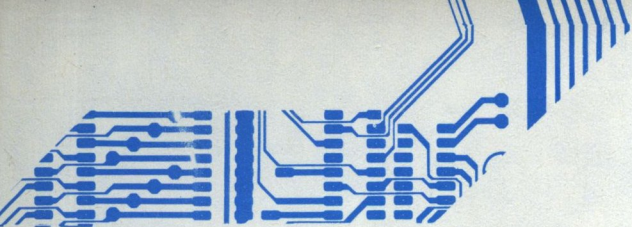
di A. Cattaneo

Questo circuito mette a disposizione 16 Kbyte effettivi, suddivisi tra due RAM, che funzionano come pseudo-ROM in quanto vengono utilizzate RAM CMOS alimentate mediante una batteria in tampone, per conservare i dati in caso di mancanza di corrente.

I possessori del computer C64, sono molti e certamente tra questi ci saranno molti principianti: ecco qualche indicazione elementare dedicata proprio a loro.
Sul lato destro del pannello posteriore del C64 si trova un connettore lineare a 44 poli in grado di accogliere una scheda, oppure un connettore per circuito stampato. Questo connettore è chiamato "porta di espansione", oppure "bus", oppure ancora "slot per schede ausiliarie". In realtà, si tratta di un accesso versatile e flessibile ai circuiti interni del computer tramite il quale potranno essere collegate ulteriori porte, banchi di memoria, strumenti di misura, decodificatori per comunicazioni, nonchè la nostra ROM fittizia. I 44 collegamenti sono suddivisi in due: dal lato superiore sono numerati da1 a 22, da sinistra a destra con la tastiera davanti ed una serie inferiore, contrassegnata dalle lettere da A a Z, escluse le lettere G, I, O, Q. La Tabella 1 e la Figura 1 mostrano a quali piedini corrispondono i diversi segnali presenti.
Per la nostra scheda da 16 Kbyte, i segnali presenti ai piedini 8 e 9 hanno un particolare significato: tramite gli ingressi GAME negato ed EXROM negato è infatti possibile influire sulla gestione degli indirizzi del C64.
L'intera gestione degli indirizzi nel C64 è a carico del circuito integrato U17, una matrice logica programmabile (PLA = Programmable Logic Array). In questo circuito integrato entrano 16 linee, tra le quali GAME negato ed EXROM negato, che permettono in teoria di ottenere 65536 combinazioni. Dal circuito integrato escono soltanto 8 linee corrispondenti a 256 combinazioni teoriche, dirette alla BASIC-ROM, alla SYSTEM-ROM, al GENERATORE DI CARATTERI, a CIA1, a CIA2 e alla RAM.
In base al rapporto tra ingressi ed uscite (256/1), si può rilevare che molte combinazioni d'ingresso non sono semplicemente possibili, oppure possono essere ripetute varie volte, se si vuole ottenere all'uscita una corretta suddivisione del campo di indirizzamento del C64. Per facilitare la comprensione di quanto appena affermato, troviamo in Figura 2 il disegno della mappa di memoria del C64. A sinistra, accanto ai singoli blocchi da 4 K, è presente la numerazione decimale ed a destra la numerazione esadecimale delle locazioni in cui è suddivisa la memoria. Il campo di memoria è rappresentato come mappato dopo l'accensione del C64, quindi come

decretato dal sistema operativo originale. Sono perciò disponibili i seguenti segnali d'ingresso PLA:

$\overline{LORAM} = 1$
$\overline{HIRAM} = 1$
$\overline{EXROM} = 1$
$\overline{GAME} = 1$
$\overline{CHAREN} = 1$

EXROM negato e GAME negato possono essere controllate attraverso la porta di espansione. Dalla Figura 2 si può anche vedere che il EXROM negato opera nel campo da 32768 a 40959

(decimali), ovvero da $8000 a $9FFF (esadecimale). Analogamente, il campo operativo di GAME negato si estende da $A000 a $BFFF. Sembra qui che tutto sia accessibile, ma non bisogna esagerare. Questo vale anche per le quattro combinazioni possibili dei livelli logici:

Fessura di espansione per cartuccia

Pin	Tipo	Pin	Tipo
22	GND	11	$\overline{ROML}$
21	CD0	10	I/02
20	CD1	9	$\overline{EXROM}$
19	CD2	8	$\overline{GAME}$
18	CD3	7	I/01
17	CD4	6	Dot Clock
16	CD5	5	$\overline{CR/W}$
15	CD6	4	$\overline{IRQ}$
14	CD7	3	+5V
13	$\overline{DMA}$	2	+5V
12	BA	1	GND

Pin	Tipo	Pin	Tipo
Z	GND	M	CA10
Y	CA0	L	CA11
X	CA1	K	CA12
W	CA2	J	CA13
V	CA3	H	CA14
U	CA4	F	CA15
T	CA5	E	S02
S	CA6	D	$\overline{NMI}$
R	CA7	C	$\overline{RESET}$
P	CA8	B	$\overline{ROMH}$
N	CA9	A	GND

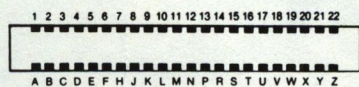



Figura 1. Pedinatura della porta di espansione presente sul lato posteriore del C64.

a) $\overline{GAME}$ = 1 $\overline{EXROM}$ = 1
b) $\overline{GAME}$ = 0 $\overline{EXROM}$ = 1
c) $\overline{GAME}$ = 1 $\overline{EXROM}$ = 0
d) $\overline{GAME}$ = 0 $\overline{EXROM}$ = 0

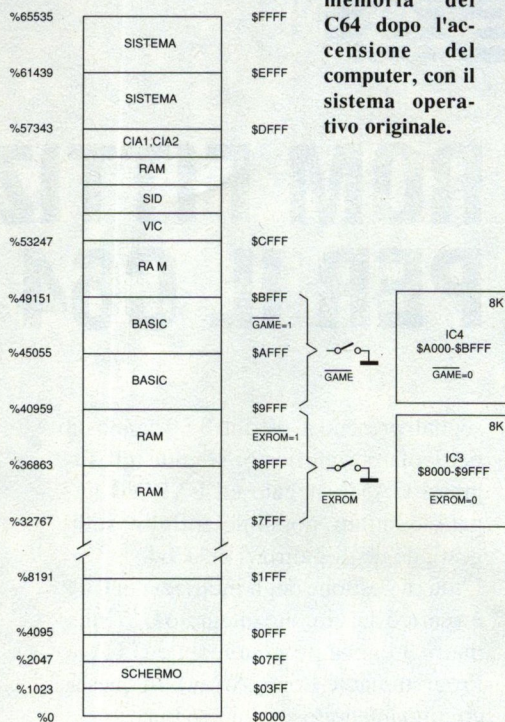
Potrete provare con calma tutte le combinazioni, il C64 non si rompe, se però date una combinazione non compatibile come può essere la b), il C64 si blocca e per continuare a godere dei suoi servizi, sarà necessario spegnerlo e poi riaccenderlo. Ancora qualche parola nei riguardi delle altre combinazioni. La a) è presente dopo l'accensione del computer. La combinazione c) verrà raggiunta azionando il commutatore S2 e la combinazione d) azionando anche il commutatore S3. Osservare ora lo schema di Figura 3. I commutatori S2 ed S3 sono posizionati in modo che non possa aver luogo la combinazione proibita b). In caso di necessità, potreste aiutarvi con il pulsante S4, che permette il reset completo del sistema operativo. Chi sviluppa programmi in linguaggio macchina, conosce bene l'utilità di questo pulsante. Dato che parliamo di commutatori, aggiungiamo che la funzione di S1 è quella di poter permettere la sola lettura, oppure la lettura/scrittura di IC3 ed IC4. E' quindi possibile utilizzare questa memoria supplementare da 16 Kbyte con alimentazione tramite batteria. Usando la combinazione c), è possibile inserire in IC3 un BASIC ampliato od un programma in linguaggio macchina. Comunque, il campo RAM da 32768 a 40959 decimali viene coperto da IC3 ed ora sono interrogate le sue locazioni di memoria.

Con i commutatori in combinazione d), potrete inserire nel C64 tutti i 16 K, utilizzando una sezione da 8 Kbyte del blocco Microsoft BASIC e una parte della RAM del C64. Se ora premete il

Figura 2. Suddivisione della memoria del C64 dopo l'accensione del computer, con il sistema operativo originale.

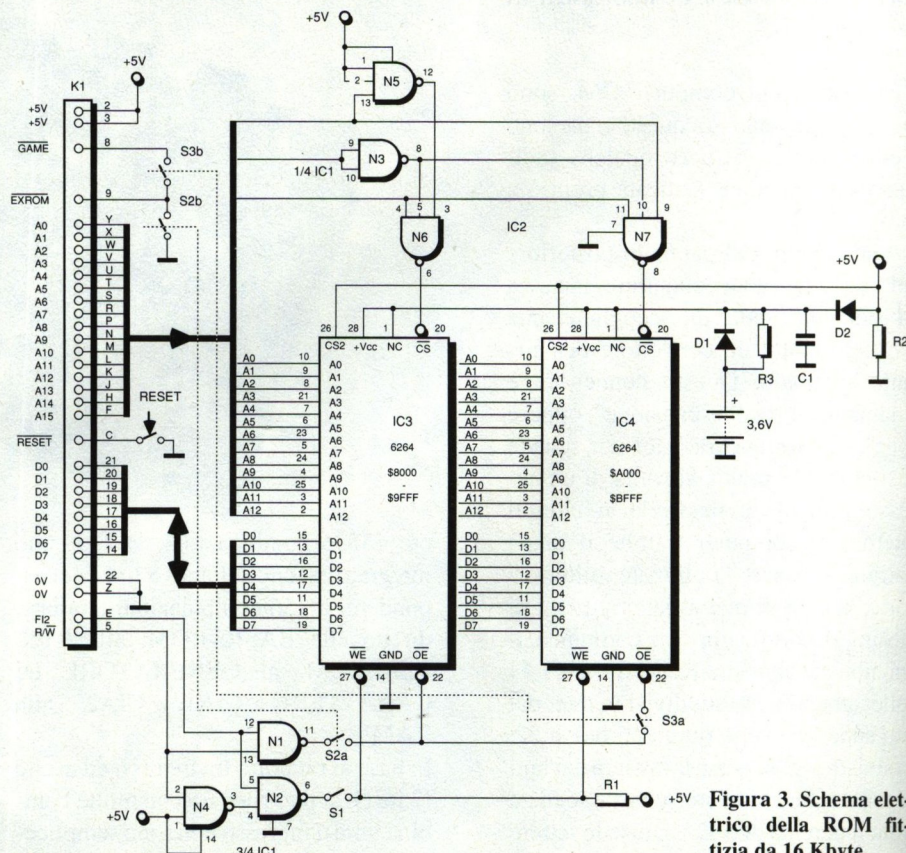



Figura 3. Schema elettrico della ROM fittizia da 16 Kbyte.

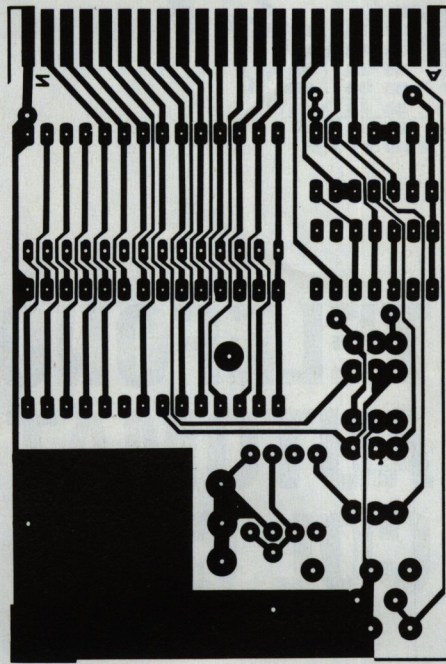
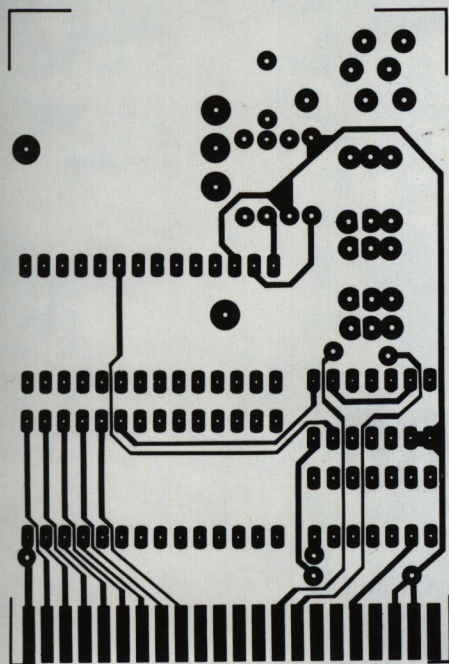



Figura 4. Piste di rame del circuito stampato a doppia faccia della scheda di estensione.

pulsante di RESET S4, in IC3 sarà presente quanto segue:

$8004	$8005	$8006	$8007	$8008
C3	C2	CD	38	30
C	B	M	8	0

(numeri esadecimali)

Verrà poi effettuato un salto all'indirizzo locato in $8000, $8001 (byte basso, byte alto) e i due byte successivi

Figura 5. Disposizione dei componenti sulla basetta.

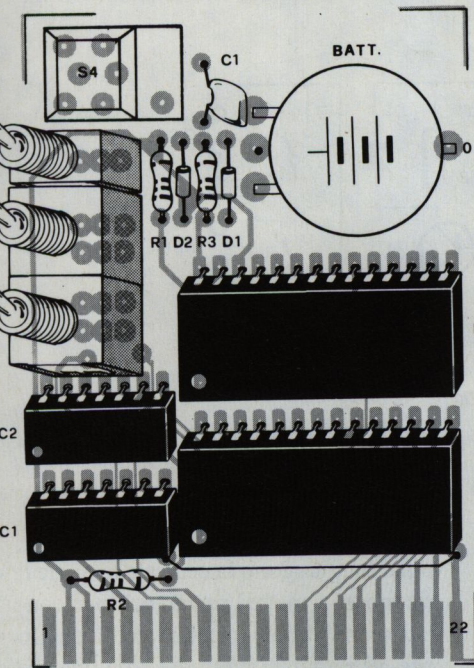
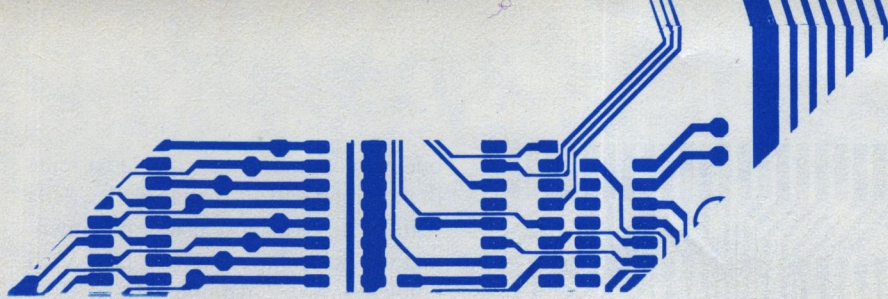
conterranno il vettore di avviamento. Qui si trovano anche i bit di arresto del C64. Uno sguardo alla traccia rame di Figura 4 e alla disposizione dei componenti di Figura 5, permetterà di constatare la notevole compattezza della scheda, dovuta anche al fatto che questa non può essere più larga della porta di espansione. Il circuito stampato è a doppia faccia e su di esso dovranno essere montati quattro circuiti integrati, quattro commutatori, tre resistori, due diodi, un condensatore, un accumulatore ed un ponticello: non occorre altro. Saldare dapprima i componenti di minore ingombro come i resistori, i diodi, il condensatore ed il ponticello, quindi passare agli zoccoli che, in questo caso, sono indispensabili. Per i commutatori S1-3 esistono versioni per circuito stampato, mentre S4 è un pulsante digitast. Come accumulatore, è bene usare un elemento a bottone da 3,6 V e terminali a linguetta per saldatura, che permette il montaggio più stabile: è tutto!

La selezione avviene bit per bit. Il resistore R1 provvede ad impedire la scrittura in corrispondenza ad un determinato stato degli ingressi WE negato. Le porte logiche N3 ed N5-7 formano, a partire dagli indirizzi A13-15, il segnale di selezione del chip. La porta logica N1 ed i commutatori S2A-S3a effettuano lo sblocco delle informazioni, in sincronismo con il segnale di clock, alle uscite dei dati del chip. La combinazione D1-R3 provvede da un lato alla conservazione dei dati in memoria con la corrente di batteria e dall'altro alla ricarica della batteria stessa con la tensione di alimentazione del computer che è di 5 V. Il condensatore C1, il diodo D2 ed il resistore R2 provvedono ad una commutazione graduale tra i due tipi di alimentazione. Per concludere, purchè l'accumulatore non venga staccato, la ROM fittizia si comporterà sempre come tale grazie alla piccolissima corrente assorbita dai circuiti integrati visto che IC1 ed IC2 sono del tipo HCT. La possibilità di conservare i dati per anni, senza doverli continuamente riscrivere, proprio come se fosse una vera ROM, rende la nostra scheda particolarmente adatta in quelle applicazioni in cui sia richiesta una grande velocità di caricamento.

Piedino	Definizione	Descrizione
1	GND	Potenziale di massa
22	GND	Potenziale di massa
A	GND	Potenziale di massa
Z	GND	Potenziale di massa
2/3	+5V (Vcc)	Positivo di alimentazione
4	$\overline{IRQ}$	Richiesta di interruzione attiva a livello "0"
5	R/$\overline{W}$	Lettura ("1") o scrittura ("0")
6	DOT CLOCK	Sincronismo del chip video
7	$\overline{I/O\ 1}$	Gestione indirizzi (uscita)
11	$\overline{ROML}$	Gestione indirizzi (uscita)
B	$\overline{ROMH}$	Gestione indirizzi (uscita)
10	$\overline{I/O2}$	Gestione indirizzi (uscita)
8	$\overline{GAME}$	Gestione indirizzi (ingresso)
9	$\overline{EXROM}$	Gestione indirizzi (ingresso)
12	BA	Validita' dati chip video
13	$\overline{DMA}$	Al pin AEC del 6510
14-21	D7-0	Bus dati del 6510
C	$\overline{RESET}$	A livello "0", inizializza il C64
D	$\overline{MNI}$	A livello "0" provoca una interruzione non mascherabile
E	FI 2	Uscita clock del 6510
F-Y	A15-0	Bus indirizzi del 6510

Tabella 1. Funzioni dei piedini della porta di espansione del C64 e relativa descrizione.

ELENCO DEI COMPONENTI

R1	**resistore da 10 kΩ 1/4 W**
R2-3	**resistori da 4,7 kΩ 1/4 W**
C1	**cond. ceramico da 100 nF**
D1-2	**diodi 1N4148**
IC1	**circuitointegrato 74HCT00**
IC2	**circuito integrato 74HCT10**
IC3-4	**RAM 6264 (Hitachi)**
S1	**interruttore unipolare**
S2-3	**deviatori bipolari**
S4	**pulsante digitast**
1	**accumulatore Ni-Cd da 3,6 V per cs**
1	**circuito stampato**

TELEFONO AUTOMATICO COL C64

di A. Cattaneo

Eccovi un utilissimo circuito studiato in origine per il C64 ma adattabile, con poche modifiche, a qualsiasi altro home computer.

Succede molto spesso per ragioni di lavoro o di urgenza di dover usare ripetutamente il telefono nel volgere di breve tempo.

Nella maggior parte dei casi, la ricerca del numero sull'agenda e la sua composizione sul disco combinatore del telefono richiede più tempo che la conversazione, senza contare eventuali errori o casi di linea occupata. Come accade da qualche tempo un pò per tutte le cose, anche in questo caso il computer può venirci in aiuto e sollevarci dall'ennesimo impegno che è appunto quello di metterci in comunicazione con l'interlocutore lontano.

Non si pensi subito al robot tuttofare capace di rispondere a tono a chi ci telefona, ma ad un più realizzabile compositore di numero telefonico gestito dal computer. Sfruttiamo quindi una macchina per mezzo di un'altra macchina, e rivolgiamo la nostra attenzione e il nostro tempo libero ad altre faccende.

La realizzazione che presentiamo, è un combinatore telefonico studiato originariamente per il C64, usabile istantaneamente sul VIC20 e, con qualche modifica, su altri computer. Bastano un semplice programma BASIC di gestione e pochissimo hardware, e il nostro C64 diventa un telefono automatico da fare invidia a quelli commerciali costosissimi. Prima di affrontare il funzionamento del dispositivo è doveroso soffermarci un pò sul modo in cui funziona un telefono e in particolare sul suo disco combinatore che chissà quante volte abbiamo azionato.

Nel momento in cui alziamo la cornetta, un contatto inserisce in parallelo alla linea telefonica un carico di circa 600 Ω costituito dalla parte fonica del ricevitore riducendo drasticamente la tensione ai capi della linea la quale scende dai 50 e più volt a circa 8 volt attivando il selettore di centrale il quale si pone in attesa degli impulsi che compongono il numero. In effetti, quando noi componiamo un numero, non facciamo altro che attivare una coppia di contatti di cui è provvisto il combinatore stesso. Uno dei due interruttori è collegato in serie alla linea e si apre sequenzialmente producendo un numero di impulsi pari a quello della cifra composta, ad eccezione dello 0 il quale genera 10 impulsi. L'altro contatto cortocircuita il ricevitore per evitare che gli stessi impulsi si presentino in cornetta sovraccaricando i timpani di chi sta telefonando. Gli impulsi raggiungono la centrale la quale li riconosce e mette l'apparecchio in contatto con l'abbonato chiamato. E' chiaro che gli impulsi inviati dal combinatore presentano caratteristiche particolari, infatti hanno un periodo di 100 ms e un

Figura 1. Schema elettrico del telefono automatico per C64.

rapporto duty-cycle di 40-60. In pratica, componendo il numero "1", invieremo alla linea un solo impulso interrompendola per 60 ms e ricollegandola per altri 40. Due cifre successive dello stesso numero devono essere distanziate da una pausa di circa 500 ms, se la tolleranza è troppa, la centrale non riesce ad interpretare il numero stesso.

L'interfaccia ha il compito di sostituire i contatti del disco, mentre la temporizzazione del segnale si ottiene via software senza troppi problemi.

Tutto questo può apparire sul subito assai complesso, ma se consultate il disegno di Figura 1, vi ricrederete all'istante. Il montaggio richiede solo due transistor oltrechè una manciata di componenti passivi.

Il circuito elettrico si compone di pochissimi componenti ed è connesso allo slot utente del C64 o del VIC 20 dal quale sono ricavati i segnali di pilotaggio e l'alimentazione. Il C64 dispone infatti, tramite lo slot utente, di 8 porte di I/O gestite da due registri alle locazioni 56577 e 56579. Il primo registro stabilisce il modo di funzionamento (input od output) di ciascuno dei terminali da PB0 a PB7 della porta utente, l'altro rileva o pilota i cambiamenti di stato logico.

In questo modo diventa molto semplice pilotare i due relè dell'interfaccia grazie ad un breve programma. Il programma va modificato nelle sue POKE qualora venisse impiegato in combinazione col VIC20 e più precisamente: la 56577 diventa 37136 e la 56579 diventa 37138.

Il montaggio del circuito non prevede difficoltà e potrà essere intrapreso anche da chi è alle prime armi; valgono comunque le solite raccomandazioni di controllare il tutto prima del collaudo finale. Il piccolo circuito stampato lo trovate in scala unitaria nella Figura 2 mentre la disposizione dei componenti è illustrata in Figura 3. Prestate attenzione alla polarità dei diodi usati, (qualche volta accade che la tacca indicante il terminale positivo non corrisponda), all'inserimento del condensatore elettrolitico e dei transistor. Il connettore 12+12 poli di collegamento con il computer è posto a cavallo del circuito stampato ed è saldato solo nella parte inferiore; superiormente dovrà essere effettuato il solo collegamento dell'alimentazione: dall'ingresso positivo al pin 2 del connettore. Il collaudo potrà essere fatto collegando al computer (spento) il circuito e caricando il programma presentato nel listato; al termine, battendo RUN, dovrete vedere funzionare i due relè. Il collegamento alla linea telefonica non è affatto difficile, se disponete di un apparecchio a spina risulterà addirittura elementare come potete vedere dalla Figura 4. Il listato non è altro che un esempio di funzionamento del circuito e non si pone altri fini se non quello di essere un esempio per lo sviluppo di programmi futuri da parte dell'utente. Una cosa è certa, che il computer può dare molto di più che non le prestazioni dettate da questo semplice programma; ad esempio è possibile l'allestimento di un'agenda che componga il numero automaticamente, oppure l'invio automatico di messaggi utilizzando un registratore un pò come una segreteria automatica.

Figura 2. Circuito stampato del combinatore visto dal lato rame in scala unitaria.

Figura 4. Collegamento da effettuare alla linea telefonica.

Figura 3. Disposizione dei componenti sulla basetta del combinatore.

```
5 REM SE IL PROGRAMMA VIENE USATO COL VI
  C20, SOSTITUIRE
6 REM L'INDIRIZZO 56577 CON 37136 E L'IN
  DIRIZZO 56579 CON 37138.
10 PRINT"{CLR}"
15 PRINT"FAI IL NUMERO,{26 SPC}OPPURE 'R
   ' PER RIATTACCARE"
20 INPUTA$
25 GOSUB35
30 GOTO1
35 IFA$="R"THENGOTO120
40 POKE56579,3
45 POKE56577,2
50 FORT=1TO200:NEXTT
55 FORA=1TOLEN(A$)
60 LETB$=MID$(A$,A,1):LETB=VAL(B$)
65 IFB=0THENLETB=10
70 FORC=1TOB
75 POKE56577,3
80 FORT=1TO35:NEXTT
85 POKE56577,2
90 FORT=1TO30:NEXTT:NEXTC
100 FORT=1TO400:NEXTT:NEXTA
110 POKE56577,0
115 RETURN
120 POKE56579,3
125 POKE56577,1
130 FORT=1TO1200:NEXTT
135 POKE56577,0
140 RETURN
```

ELENCO DEI COMPONENTI

R1-2	**resistore da 4,7 kΩ**
R3-4	**resistore da 2,2 kΩ**
C1	**cond. elettr. da 470 μF 16Vl**
D1-2	**diodi 1N4148**
T1-2	**transistori BC107**
RL1-2	**relè 6V 1 scambio**

MINI ROBOT

di M.Bovio

Con questo breve articolo, vogliamo polarizzare l'attenzione di tutti coloro che sono appassionati di automazione computerizzata, presentando un semplice supporto modulare particolarmente adatto per semplici applicazioni. E' appunto la sua semplicità che ne suggerisce un impiego didattico nelle scuole e pratico nei laboratori, per allestire apparecchiature di prova.

Questo piccolo kit, se corredato di buoni controlli meccanici, può comunque trovar posto anche in impianti di una certa importanza e potrete divertirvi moltissimo costruendo modelli di robot senza doverli necessariamente collegare ad un qualsiasi computer, infatti il nostro circuito può funzionare con un sistema di interruttori interconnessi, fotocellule e, naturalmente con tutti i controlli manuali. I due motori vanno dotati ciascuno di un potenziometro il cui perno sia coassiale con l'albero del motore stesso in modo da informare il computer (attraverso un convertitore A/D) di quale posizione assumano di volta in volta i bracci, le leve e le camme sul modello che si sta pilotando. Provate a montare i motorini su qualche modello LEGO o MECCANO e potrete facilmente apprezzare quanto siano originali. I motori possono essere piccoli ma, nello stesso tempo, estremamente potenti e devono girare benissimo con tensioni anche di 3 V. Non alimentate però i motori attraverso i 5 V del vostro microcomputer, perchè la potenza sarebbe del tutto insufficiente. E' necessario un alimentatore di rete che eroghi una tensione di +8 V, con una corrente di 0,5 A. La combinazione motore-riduttore è particolarmente efficiente nelle disposizioni di movimentazione a cremagliera e pignone (ad esempio per gli spostamenti lungo l'asse X della penna di un plotter).

Figura 1. Schema elettrico del circuito di controllo per due motori.

Per generare sequenze di movimenti, è necessario un programma, ma prima di questo ci occorre un computer con la relativa interfaccia e poichè non tutti i computer hanno lo stesso sitema operativo, è bene munirsi di una apposita interfaccia. Comunque, se possedete una scheda I/O contenente alcuni relè pilotati dal vostro computer, non avrete difficoltà a collegarla al nostro circuito. Se però possedete soltanto una scheda I/O od una porta di computer che fornisca, per esempio 8 uscite TTL, dovrete bufferizzare senza risparmio i segnali I/O prima che raggiungano i motori, anche se questi sono alimentati a 5 V. Lo shema elettrico del circuito di pilotaggio dei due motori lo si può trovare in Figura 1. Si tratta di un circuito perfettamente sicuro, che utilizza il BIT 0 ed il BIT 1 della vostra porta d'uscita (potrebbe essere la porta Centronics) per far girare in avanti od all'indietro un motore. Per pilotare il secondo motore, sarà necessario un circuito identico, che utilizza però il BIT 2 ed il BIT 3 della porta stessa.

Il circuito è "sicuro" in tre modi diversi. In primo luogo, eventuali guai relativi al cablaggio sul modello metteranno in pericolo soltanto i poco costosi circuiti integrati IC1 ed IC2: microcomputer non subirà danni. In secondo luogo, l'alimentazione per l'interfaccia ed i motori viene prelevata da una sorgente esterna e non può sovraccaricare l'alimentatore a 5 V del microcomputer. Infine, il blocco del software non costringerà i motori a fare cose pazze, come tentare di girare contemporaneamente in due direzioni.

L'interfaccia potrà essere costruita

agevolmente su una basettina preforata per prototipi oppure realizzando il circuito stampato di cui la traccia rame in Figura 2. In Figura 3 è invece visibile la disposizione dei componenti. La tensione d'ingresso del regolatore non dovrebbe essere inferiore agli 8 V c.c., perchè il 7805 necessita di una differ-

Figura 2. Circuito stampato del controllo visto dal lato rame in scala unitaria.

enza di almeno 3 V tra alimentazione ed uscita per funzionare correttamente. L'alimentazione del motore potrà essere derivata dagli stessi +8 V, oppure potrà essere una qualsiasi altra tensione, compresa tra +3,5 e 9 V. L'alimentatore da 8V sopra ricordato, si adatta perfettamente. Accertarsi che i diodi D1-D4 siano montati con la corretta polarità, altrimenti i transistori verranno distrutti dalla tensione inversa proveniente dai motori quando questi si trovano in decelerazione. I transistori sono del tipo ad alta potenza, ma non richiedono dissipatori termici.

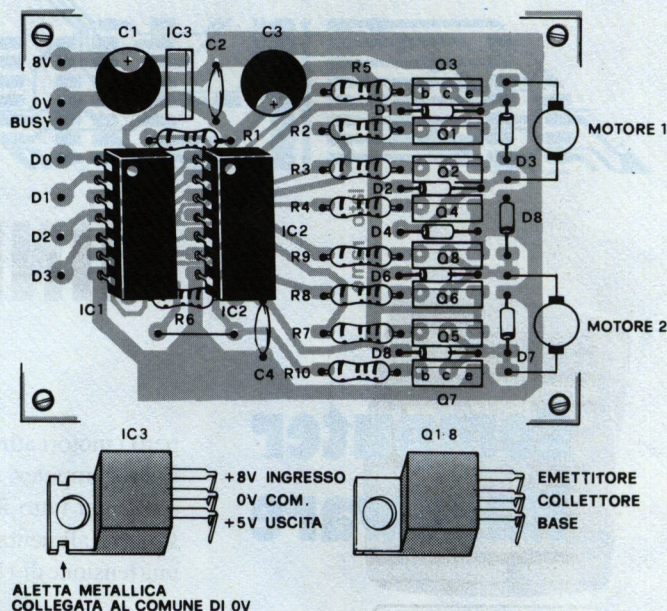



Figura 3. Disposizione dei componenti sulla basetta stampata.

Il software

Supponendo che l'indirizzo di porta del computer sia caricato nella variabile P, per far partire il motore 1 in avanti, bisognerà dare:

POKE P,1 (se non si dispone di un comando OUT)

oppure

OUT P,1

Per arrestare la marcia si dovrà dare:

POKE (oppure OUT) P,0

Allacciando l'interfaccia alla porta per stampante Centronics (se il computer la prevede) e mantenendo a livello BASSO la linea BUSY, potrete usare:

LPRINT CHR$(1) e LPRINT CHR$(0)

I comandi citati pilotano il giusto motore nella giusta direzione, a patto che i collegamenti alla presa del modello siano corretti. Il codice per l'azionamento dei due motori in un senso o nell'altro, è indicato in Tabella 1.
Stendendo appropriati diagrammi di flusso, sarà possibile scrivere programmi di controllo per robot in qualsiasi linguaggio che non sia il BASIC, per cui sarà possibile ricorrere al FORTH o al PASCAL, con tutti i vantaggi che questi linguaggi comportano. Con un'interfaccia più sofisticata si potrà anche leggere la tensione fornita dai potenziometri, mentre una programmazione più complessa permetterà al robot di imparare, memorizzare ed eseguire sequenze controllate e ripetitive di diversi movimenti.

Fine operazione	00000000	0
Motore 1 acceso, avanti	00000001	1
Motore 1 acceso, indietro	00000011	3
Motore 2 acceso, avanti	00000100	4
Motore 2 acceso, indietro	00001100	12
Esistono naturalmente anche altre combinazioni valide, per esempio:		
Entrambi i motori avanti	00000101	5

Tabella 1. Valori da trasmettere alla porta.

ELENCO DEI COMPONENTI

R1-8	resistori da 100 Ω, 0,25 W
C1	cond. elettr. DA 470 µF 16 Vl
C2-4	cond. ceramici a disco da 100 nF
C3	cond. elettr. da 100 µF 10 Vl
IC1	circuito integrato 74LS08
IC2	circuito integrato 74LS06
IC3	circuito integrato 7805
Q1-3-5-7	transistori TIP32
Q2-4-6-8	transistori TIP31
D1/8	diodi 1N4148
2	zoccoli per c.i. a 14 pin

LETTURA COMPUTERIZZATA DELLA FREQUENZA

di M. Anticoli

E' questo un circuito di lettura digitale della frequenza per ricevitori, adattabile non solo al modello Yaesu sotto riportato, ma anche a qualsiasi altro ricevitore.
Per anni abbiamo desiderato un display digitale sul nostro ricevitore a copertura generale Yaesu FRG-7, per migliorare la precisione delle annotazioni di ascolto sulle onde corte. La nostra esperienza con questo ricevitore aveva infatti dimostrato che la taratura della scala meccanica non è precisa nell'intera banda.
Prendendo spunto dal progetto di un frequenzimetro, abbiamo gettato le basi per questo display digitale.

Speciale radioamatori e CB

All'interno del ricevitore

Il ricevitore FRG-7 utilizza un sistema di sintonia VFO a scala principale, la cui frequenza varia da 3455 a 2455 kHz, per sintonizzarsi lungo i MHz della banda desiderata. Il VFO oscilla a 3455 kHz all'estremo basso della scala ed a 2455 kHz all'estremo alto. La sintonia avviene all'indietro. Questa tecnica, come parecchie altre, è comunemente utilizzata nei moderni ricevitori. Misurando la frequenza del VFO e sottraendola da 3455 kHz, si può calcolare la frequenza ricevuta. Anche se può essere utilizzato hardware dedicato per ricavare questo numero e visualizzarlo, abbiamo preferito lasciare questa fatica al nostro C64. Una riga o due di BASIC in più permettono di entrare ed uscire facilmente dal programma e perciò il computer potrà essere utilizzato anche per altri compiti.
Per collegare il ricevitore al computer, è necessario un circuito di interfaccia. I componenti elettronici ed il connettore 12/24 non costano molto e aggiungendo un piccolo contenitore, il circuito stampato e pochi altri componenti, il prezzo rimane alla portata di tutti.
Lo schema dell'interfaccia è illustrato in Figura 1. La prima parte (Q1, Q2 e Q3) è accoppiata all'amplificatore buffer del VFO contenuto nel ricevitore. Nell'FRG-7, questo punto (TP-404) è ben indicato sul circuito stampato della media e della bassa frequenza. I due circuiti integrati 7490 sono cablati in modo da dividere per 10 la frequenza del VFO. Essi correggono la forma dell'onda e garantiscono la compatibilità TTL con il C64. Servono inoltre ad effettuare un primo abbassamento della frequenza (prescaling), in modo da non eccedere il limite di 65535 eventi, sopportabile

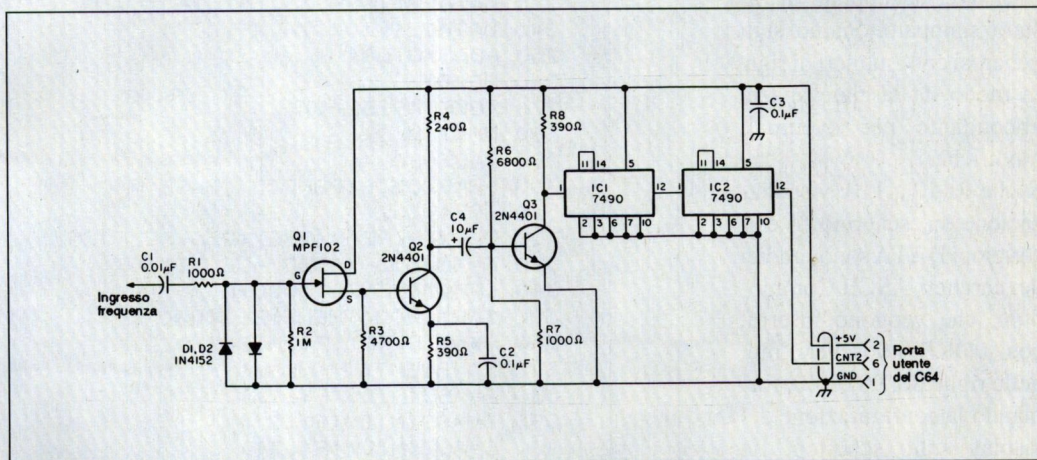



Figura 1. Schema elettrico dell'interfaccia per la lettura della frequenza di un ricevitore.

dal registro d'ingresso del computer.
La disposizione circuitale non è critica. Abbiamo usato una piccola basetta universale inserita in una scatola di alluminio e poi installata all'interno dello stesso ricevitore. Il collegamento al VFO verrà effettuato con un corto spezzone di cavo schermato. Il cavo diretto al computer è a due conduttori (+5 V e CNT2), con lo schermo collegato a massa.

Il programma

La descrizione del programma che segue, fornirà spiegazioni sufficienti a comprendere il funzionamento, permettendovi anche di personalizzarlo a seconda delle vostre particolari necessità. Abbiamo rinunciato ad usare le istruzioni REM, ricorrendo invece a spiegazioni separate, più estensive, accoppiate al listato stampato. Siamo stati generosi con i numeri di riga, in modo da lasciare spazio abbondante per eventuali modifiche.
Righe da 10 a 110: visualizzazione su schermo. Le istruzioni DATA sono i codici dei caratteri ASCII Commodore, che vengono inseriti, con POKE alloscher- mo, nelle righe 80, 90 e 100, formando la configurazione del display sullo schermo. La riga 70 cancella e libera lo schermo mentre viene costruito il display. La riga 110 riattiva lo schermo.
Righe 120-320: routine di conteggio.
La routine di conteggio è un programma in linguaggio macchina. Le righe da 120 a 240 contengono il vero e proprio programma in l.m., mentre le righe 250-280 trasferiscono il programma nella memoria, a cominciare dall'indirizzo 49152. La riga 290 regola ad un secondo la base dei tempi, effettua il POKE in memoria del valore (riga 300) ad avvia il contatore con il comando SYS nella riga 310.
Il contatore viene letto nella riga 320.

```
10 REM COMMODORE 64 DIGITAL FREQUENCY DISPLAY
20 DATA236,226,226,226,226.226
30 DATA226,226,226,226,226,226,251
40 DATA97,32,32,32,32,32,32,32,11,8,26,225
50 DATA252,98,98,98,98,98,98,98,98
60 DATA98,98,98,254
70 PRINTCHR$(147):POKE53281,14
80 FORZ=0TO12:READX:POKE1197+Z,X:NEXTZ
90 FORZ=0TO12:READX:POKE1237+Z,X:NEXTZ
100 FORZ=0TO12:READX:POKE1277+Z,X:NEXTZ
110 POKE53281,1
120 DATA8,120,169,20,141
130 DATA20,3,169,192,141
140 DATA21,3,169,3,141
150 DATA13,221,40,88,96
160 DATA72,198,255,240,4
170 DATA104,76,49,234,165
180 DATA251,133,255,169,0
190 DATA141,14,221,173,5
200 DATA221,133,252,173,4
210 DATA221,133,253,169,255
220 DATA141,5,221,141,4
230 DATA221,169,33,141,14
240 DATA221,104,76,49,234
250 FORI=0TO64
260 READX
270 POKE(49152+I),X
280 NEXTI
290 BASE=60
300 POKE255,0:POKE251,BASE
310 SYS49152
320 COUNT=(PEEK(252)*256+PEEK(253))
330 F=(34550-(65535-COUNT))
340 IFF<0THENF=10000+F
350 IFF>10000THENF=F-10000
360 F$=STR$(F):F$=MID$(F$,2)
370 F$=RIGHT$("      "+F$,5)
380 A=ASC(MID$(F$,1))
390 B=ASC(MID$(F$,2))
400 C=ASC(MID$(F$,3))
410 D=ASC(MID$(F$,4))
420 E=ASC(MID$(F$,5))
430 POKE1238,A:POKE1239,B:POKE1240,C:POKE1241,D
440 POKE1242,46:POKE1243,E:POKE1244,48
450 FORW=1TO1000:NEXTW
460 GETA$:IFA$=CHR$(13)GOTO480
470 GOTO320
480 PRINTCHR$(147):SYS58235
```

Righe 330-350: calcolo della frequenza
La riga 330 calcola la frequenza del ricevitore. Il numero effettivo di conteggi è 65535. Questo numero viene poi sottratto dalla frequenza corrispondente all'estremo inferiore della scala (frequenza del VFO di 3455 kHz), per calcolare il valore della frequenza da visualizzare.
Le righe 340 e 350 permettono di regolare il numero per i casi in cui la scala principale sia maggiore o minore dalla banda scelta.
Righe 360-370: formato di visualizzazione della frequenza. La riga 360 converte il numero relativo alla frequenza in una STRinga che permette la facile formattazione della frequenza da visualizzare sullo schermo. F$=MID$(F$,2) cancella gli spazi iniziali che il BASIC inserisce automaticamente per il segno del numero (+ o -). La riga 370 aggiunge spazi iniziali alla stringa e poi limita la lunghezza della

stessa a cinque spazi a partire da destra. Le righe da 380 a 420 assegnano le variabili a ciascuno dei cinque caratteri della stringa.

Le righe 430 e 440 effettuano il POKE della frequenza sullo schermo. Osservate che questa istruzione fissa la posizione del punto decimale e lo zero iniziale.

La riga 450 è necessaria per rallentare il programma mentre il contatore viene aggiornato. La riga 460 permette di uscire dal programma e di ritornare al BASIC premendo il tasto RETURN, che rinvia il programma alla riga 480 per cancellare lo schermo ed eseguire un'inizializzazione a caldo del computer. La riga 470 fa ciclare il programma, per aggiornare il display se non è stato premuto il tasto RETURN.

Modifiche al programma

La base dei tempi per il contatore utilizza le interruzioni basate sul clock interno del C64, che avvengono ogni sessantesimo di secondo (in gergo "jiffie").

La base dei tempi del frequenzimetro può essere programmata in multipli di questa grandezza.

Questa operazione viene effettuata nella riga 290, dove abbiamo scelto 60 jiffie, equivalenti ad 1 s. Se modificate la base dei tempi, accertatevi di modificare opportunamente anche il calcolo della frequenza della riga 330. Se scegliete, ad esempio, una base dei tempi di 6 jiffie (1/10 di s), il risultato del calcolo della frequenza dovrà essere moltiplicato per 10. Un modo per ottenere questo risutato, sarebbe di aggiungere la riga 335: F=F*10.

Il programma qui stampato è solo una parte di quello che abbiamo usato realmente. Definendo le altre condizioni (comandi da tastiera) nell'istruzione GET della riga 460, potrete saltare ad altri programmi, per esempio un logbook, un RTTY/CW, eccetera.

Abbiamo usato questa tecnica per ridefinire il calcolo della frequenza, usando il convertitore VLF nel nostro ricevitore.

Il nostro convertitore esplora le frequenze da 10 a 500 kHz sintonizzando la sezione compresa tra 3510 e 4000 kHz della scala del ricevitore.

Abbiamo usato la barra spazio per ordinare al programma di sottrarre altri 500 kHz dalla frequenza del VFO, quando si utilizza il convertitore VLF, e di visualizzare un messaggio, sul fondo dello schermo, per rammentare che è stata scelta questa funzione. Le relative tecniche di programmazione vanno oltre gli scopi di questo articolo.

Se desiderate un display digitale che segua i movimenti dell'indice della scala con una rapidità pari a quella delle vostre manovre, questo progetto non sarà adeguato alle vostre aspettative. Se invece siete disposti ad attendere un paio di secondi la stabilizzazione del display, ogni volta che manovrerete la manopola di sintonia del ricevitore, vi troverete soddisfatti.

Abbiamo effettuato un controllo per confronto con le frequenze note di emittenti commerciali e di servizi fissi ed abbiamo constatato una precisione compresa nei 200 Hz. La graduazione della scala del ricevitore ha invece una tolleranza di 5 kHz al centro, ma può arrivare a 30 kHz ad entrambi gli estremi.